Modeling Trust in Human Conversation

by

Catherine Miller

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of
[M. Eng.]
Master of Science in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
[June 2006]
May 2006

Author .....
Department of Electrical Engineering and Computer Science
May 22, 2006

Certified by .....
Patrick H. Winston
Ford Professor of Artificial Intelligence and Computer Science
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Theses

Modeling Trust in Human Conversation

by

Catherine Miller

Submitted to the Department of Electrical Engineering and Computer Science
on May 22, 2006, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Science and Engineering

Abstract

If we are ever to have intelligent systems, they will need memory. Memory is the core of learning; intelligence is about entering, extracting, and synthesizing its contents. What makes the memory problem difficult is that memory is not a simple collection of facts. The how and why of where those facts were acquired is a key part of how they are internalized and used later.

As a step towards solving this large and difficult problem, I have focused on how people learn to trust each other when they have a conversation. The model I have created represents people as sets of self-organizing maps; each has a map to represent his own beliefs, and a map to represent what he thinks of another person. Beliefs are in this model restricted to likes and dislikes, across a wide range of topics. In this thesis I describe the program implemented in Java to test this model. The model has been tested on four different kinds of conversations, with the topics of animals and cars, to determine whether its behavior looks reasonable to a human observer.

In this work I show how a simple, natural model can closely approximate human behavior, without need for tweaking parameters.

Thesis Supervisor: Patrick H. Winston
Title: Ford Professor of Artificial Intelligence and Computer Science

Acknowledgments

I owe both the form and content of my thesis to Patrick Winston, who introduced me to AI, taught me how to present my ideas, and gave me both freedom and advice when needed. Without him, this thesis would be forty-three blank pages.

I owe the current version of the model to the members of the Genesis Project research group who showed me the flaws in earlier ones, and gave suggestions on how to fix them. I also owe particular thanks to Mark Finlayson and Keith Bonawitz for helping navigate the murky realm of the Bridge system and Eclipse; both of them spent a considerable amount of time making my slow computer behave.

The user interface that made this thesis so easy to use would not have been possible without Amanda Seybold and Amanda Smith, who were willing to use my thesis as a final project subject when it had just begun to take form.

I owe thanks also to Reid Barton for proofreading at the last minute, testing the user interface, and being willing to help when I was in need of it. I owe Dan Katz for his support when the amount of work left to do seemed insurmountable.

I also owe a debt to all the teachers, professors, TAs and friends who have encouraged me, inspired me, and believed in me. I have been fortunate in finding role models when I most needed them, and friends where I least expected them; without them I would not be here.

Contents

List of Figures

Chapter 1

Introduction

If we are to ever be able to build an intelligent system, that system must be able to learn from its experiences. True learning does not happen when new facts are dryly entered into a memory bank; learning happens through experience. Every time a new fact is acquired, there is some context in which it is learned: the location, the person speaking, the emotional state of the learner. The problem of context is vast: humans have at least five different kinds of perception, each of which has its own particular quirks of understanding. Furthermore, there are contexts which we might call "emotional"—a beloved teacher might make a student love math, while a hated teacher might lead the student to despise equations forever.

One of the important questions in any context is whether or not trust exists. It is hard for humans to learn if they don't trust their environment and their teachers. Thus an important question is, how do we learn to trust? Trusting in parents is a genetic predisposition for young children, and for many any person in a position of authority is automatically trusted. This leads to other questions about human behavior. Why does propaganda work? Why should an advertisement on television or in a newspaper ever cause someone to vote Republican or buy a soda? Why do people sometimes believe the lies of friends who have lied to them many times before. Why do so many people believe politicians, even when the facts are contradictory? Why do others refuse to believe them no matter how much evidence there is?

In order to have a reasonable problem to attack, I choose one very specific kind

of experience and build a system that can understand interactions involving that experience within a larger context. Conversation provides that experience. If we consider only the words spoken, without considering visual cues or tone of voice, a tractable problem reveals itself. Whenever we have a conversation with another human being, what we take away from that conversation is not just the sum of the statements spoken. If some stranger approached you and claimed to have seen an elephant flying towards a tree, you would probably dismiss him as mildly insane, and the statement as being produced by a deranged mind and invalid. Why should such a statement be dismissed?

There are a number of reasons. First, a stranger has no credibility: you two have no history, so you trust him only as much as any other stranger, and your perception of him will change quickly based on the first few things he says. If instead a close friend told you that he had see an elephant in flight, you might discount the statement, but you probably don't instantly assume the friend is deranged. You might lose a little bit of confidence in him, but the reaction isn't as strong.

Why do you dismiss the statement no matter who says it? Because all your experience leads you to believe that elephants simply do not fly, no matter what the circumstance. Most people continue to believe what they already believe, even in the face of evidence to the contrary. Perhaps if everyone you know began talking of flying elephants, you might start to believe in them.

I have a built a system that begins to model these complex interactions within the mind.

Chapter 2

Background

2.1 K-lines

In a 1979 A.I. memo called *K-lines: A Theory of Memory*, and later with his 1985 book *Society of Mind*, Marvin Minsky proposes a theory of memory that in many ways satisfies my criteria. Minksy suggests that there are many agents in our minds that may or may not get activated when an event occurs. When a set of agents is triggered, a K-line connecting them is formed, which binds them together as a memory. With this theory, all memory is formed of experience, and memories are referenced when a state close to the same K-line structure is reached. While this theory satisfies the need to have memories in context, it goes too far; K-lines make it unclear how something could just be remembered as a fact and not an experience. Furthermore, no implementation details have been given, and in twenty-five years no one has claimed to have implemented it.

2.2 Threads

Also in 1979 Lucia Vaina and Richard Greenblatt proposed the use of thread memory: a kind of memory in which words are stored as threads that contain the word's superclasses. In essence, each word exists as a linear descendant of other words. For example BANANA might be BANANA → THING → LIVING-THING → FRUIT →

BANANA. Any word might have multiple threads associated with it, to represent its various characteristics. Threads allow a form of distance metric between words. Any two words are connected by tracing up the thread of one to a fork point, and down the thread of another. This memory structure is used to underpin the key model in this thesis.

2.3 Self-Organizing Maps

In order to organize my conversation memory, I employ Teuvo Kohonen's self-organizing map (SOM). The self-organizing map has a number of properties that make it ideal for this application. A SOM is a tool for bringing out the inherent regularity of a set of objects. The traditional demonstration is with colors: when a SOM is trained with colors, clusters of similar colors emerge, sometimes in the form of a rainbow, depending upon implementation. There is no need to specify the number of clusters, only two things are needed—a distance metric for the objects to be trained, and some way of making one object "more similar" to another object.

The basic algorithm assumes some starting grid of nodes of size $n \times n$, and some training set:

- Given a an input x, find the cell in the map c_i such that the distance between x and the contents of c_i is minimized for all $i, 1 \leq i \leq n^2$. "Distance" may be subjectively defined, but often simple Euclidean distance is sufficient.
- Find the set of neighboring cells of C_i that are within some radius r of c_i.
- For each member of C_i, including the winning cell c_i, make that cell "more like" x, based on its distance from c_i. Cells not within this neighborhood are not altered.

This algorithm is repeated for every element of the training set. In the general case there is some learning rate α that effects the extent to which cells are made "more like" inputs, and which might decrease over time. Because in this model SOMs are

used more as a self-organizing data structure than a clustering algorithm, and there is no set size of the inputs, this learning rate is omitted in my implementation of the algorithm.

It is worth noting that when run repetitively on a finite training set, SOMs are guaranteed to converge to a point where the cells fluctuate a distance less than ϵ as the number of cycles approaches infinity. Because in my work there is no predetermined training set—the "input" is not finite –there is not an expected outcome to which the maps converge.

2.4 Previous Work

In his 2003 Master's thesis, Stephen Larson showed that a self-organizing map can be used for symbol grounding. The system can assign meaning to symbols just by interacting in a Blocks world, without the implementer needing to step in. The fact that a SOM could be used to model how people assign meaning has been the basis and inspiration for other such work.

One attempt to use natural language and self-organizing maps as a memory system has already been pioneered by Seth Tardiff in 2004. He used a SOM, in conjunction with the BridgeSpeak parser developed by the Genesis group at MIT, to create a memory map which could understand when events were surprising—an elephant flying towards a tree, for example. My system does more than understand that something is surprising; it filters information based on past experience, specific to the individual providing the information.

Chapter 3

Overview of the Model

The key feature of my proposed model of conversation is that every human has some model of every person they've met. I have some image of my father, which is different from his mother's view of him, neither of which is probably an accurate picture of what he is. I form expectations based on my view of him, and how I internalize what he says to me depends upon that view.

To model a human being, I use one SOM to represent likes and dislikes in the world, and another for each other person that human has had interactions with.

This is the model of the inside of one person's head, not a model of the universe. The SOMs depicted are a subjective view of each person, not an accurate reflection of the world. The idea is that each of us carry inside of us opinions of everyone we have ever interacted with. These opinions help us to determine how to behave should we encounter the person again, and whether or not to believe what we're told.

In order to describe how the process is supposed to work, imagine that JOE and ME are talking and JOE says "Cats can't fly." ME does three things: he checks to see how much he trusts JOE in general, how much he trusts JOE about cats, and what he already believes about cats and flying. There should also be some parameter to represent how much weight ME gives to his own experience; some people are more gullible than others, and more likely to disbelieve their own experience. These values are then averaged, and the averaged truth value used to train ME's map of JOE, and ME's own beliefs.

The process is bootstrapping in a sense: JOE says things that ME believes, so ME trusts JOE more, so the more JOE says, the more ME agrees with him. This is the pattern of human behavior; it requires some amount of conversation to trust someone at all, but then one is inclined to believe the other person until they say something that seems implausible. Consider advertisements or political propaganda—they are premised upon the idea that people's opinions can be changed simply by telling them something frequently.

The nodes in the self-organizing maps have two components: a "content," provided by BridgeSpeak, and a "truth value." The truth value is a number representing how much the content is believed. This truth value is how trust is represented, and the number used to calculate how much a new incoming statement is believed.

Chapter 4

Architecture and Implementation

4.1 Architecture

4.1.1 Model of a Person

A person begins life as a single self-organizing map. A person may be randomly created, or he may be meticulously specified, but in principle he is a single SOM of arbitrary size, with likes and dislikes. Every node in this SOM has a statement of the form "I like X" or "I dislike X", and a percentage value that indicates how strong the preference is. These percentages can be thought of as "belief" values, and this map will be referred to as the SELF map: the map of personal beliefs.

As soon as two people meet for the first time, new maps are created. Each person gets a map that represents how much they trust the other ("OTHER" maps). In principle these maps start empty, though for practical purposes in this simulation they are randomized. As more people meet, more maps are created. These maps have the same format as the SELF maps: nodes contain statements of likes or dislikes, and percentage values. Here the percentage values are interpreted differently: they represent how much the map subject is trusted on the topic. So for example, if Zorba has a map of Bob with a node that says "I like beets, 78%", the interpretation is that Zorba believes that Bob likes beets, and trusts him a lot on the topic of beets. Essentially, Zorba is willing to give Bob's opinion on beets a lot of weight when

deciding what he himself feels about them. The percentage 78 also indicates how strongly Zorba believes that Bob likes beets—how long it would take to convince him that Bob's opinion is the opposite. The larger the number, the more firm Zorba is on his view of Bob.

4.1.2 Model of a Conversation

Once we have people, the most important thing is for them to be able to talk to each other. In this simulation, people are capable of only talking to one other person at a time, but two different people can have a common acquaintance. In other words, everyone lives in the same community, but the only way to interact is in pairs.

When people talk, they are able to talk only about their likes and dislikes. So for example, Bob may continue his trend of expressing food tastes and say "I like radishes" to Zorba. Zorba then incorporates this new information into his own world view.

He does this by finding the best match for the topic "radishes" in both his SELF map and his map of Bob. Call the result from SELF map S and the result from Bob's map B. There are several relevant variables:

```
S.trust
B.trust
L = 1 if the statement indicated like, and -1 if dislike
```

For the purposes of this algorithm, if cell X contains a "dislike" statement, then X.trust is negative.

There are two levels of blending: high and low. For "high" blending, which happens to the best matched node, the topics are blended by moving along the thread by three steps, as described in the next section. The new trust value for S is determined by:

$$\frac{2 * \text{B.trust} + \text{L}}{21} + (1 - \frac{|2 * \text{B.trust} + \text{L}|}{21}) * \text{S.trust}$$

The intuition behind this formula is that how beliefs change should depend heavily on the prior belief, and then upon the extent to which Bob is currently trusted on this topic, as well as whether what he is saying contradicts what he has said before. The first term corresponds to how trustworthy and consistent Bob is on the topic, and the second term to what the current belief is.

The new trust value for B is:

$$\frac{\text{L}}{5} + \frac{3 * \text{B.trust}}{5} + \frac{\text{S.trust}}{5}$$

Here the intuition is that the statement itself and the prior trust value are most significant (and trust should be lower if they contradict), but there is a "disbelief" element that makes it harder to believe things that contradict one's own world view.

Low levels of blending happen to the eight neighbors of S and B. "Low" blending alters topics by moving two steps along the threads. New trust values are computed using denominators of 30 and 7 instead of 21 and 5. These numbers were somewhat arbitrarily chosen; they produce reasonable behavior, but so do most other choices. While these constants affect the specifics of how fast the algorithm works, changing the numbers should correspond to the variance between people, not to the difference between what is human and what is not. Tweaking values changes the minute details of behavior, not the underlying patterns.

Here are some examples of the effects of blending given different starting states. These examples look only at the "winning" nodes in both the SELF and OTHER maps, so they show a high level of blending. In all cases the scenario is that the other person said "I like carrots" and the winning nodes in both the SELF and OTHER maps have the topic potatoes.

1. SELF strongly likes potatoes, believes that OTHER likes potatoes, and strongly trusts him about potatoes.

SELF	like potatoes .9	"I like carrots" →	like potatoes .913
OTHER	like potatoes .9	"I like carrots" →	like potatoes .92

2. SELF strongly likes potatoes, believes that OTHER doesn't like potatoes, and strongly trusts him about potatoes.

SELF	like potatoes .9	"I like carrots" →	like potatoes .828
OTHER	dislike potatoes .9	"I like carrots" →	dislike potatoes .16

3. SELF strongly dislikes potatoes, believes that OTHER likes potatoes, and strongly trusts him about potatoes.

SELF	dislike potatoes .9	"I like carrots" →	dislike potatoes .647
OTHER	like potatoes .9	"I like carrots" →	like potatoes .56

4. SELF strongly dislikes potatoes, believes that OTHER doesn't like potatoes, and strongly trusts him about potatoes.

SELF	dislike potatoes .9	"I like carrots" →	dislike potatoes .904
OTHER	dislike potatoes .9	"I like carrots" →	dislike potatoes .52

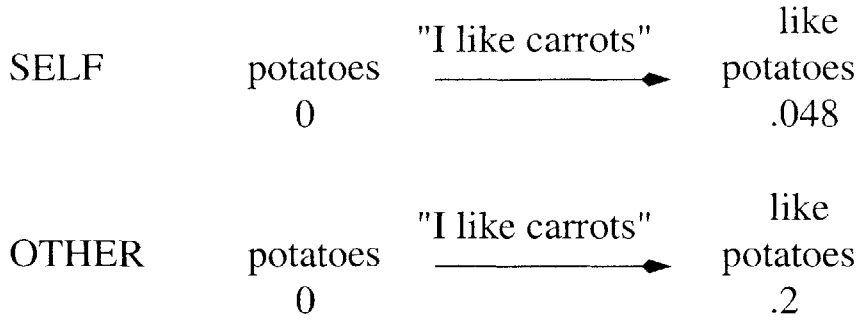
5. SELF strongly likes potatoes, has no notion of whether OTHER likes or dislikes them, and does not trust OTHER at all about potatoes.

SELF	like potatoes .9	"I like carrots" →	like potatoes .905
OTHER	potatoes 0	"I like carrots" →	like potatoes .38

6. SELF has no opinion on potatoes, believes that OTHER likes potatoes, and strongly trusts him about potatoes.

SELF	potatoes 0	"I like carrots" →	like potatoes .133
OTHER	like potatoes .9	"I like carrots" →	like potatoes .74

7. SELF has no opinion on potatoes, has no notion of whether OTHER likes or dislikes them, and does not trust OTHER at all about potatoes.



4.1.3 SOM Comparisons

In order to update the self-organizing maps, it needs to be possible to compare two words. Retrieval is always on the topic of the sentence alone, so "I like zoos" is the same as "I dislike zoos" for this purpose: the important part is to get a similar topic, not a similar opinion.

These comparisons are done using threads obtained from WordNet via the Bridge system. Each word in the dictionary actually has a number of different threads associated with it: "Bill Clinton" may be on a thread that contains "politician" and "president" and also another that says "father." For the purposes of this demonstration I look only at the first thread that WordNet offers–usually the most common sense.

To compare two words I take both of their threads and look for common words lowest down. For example, to compare "dog" and "cat", their threads are:

```
<thread>entity object living_thing organism animal chordate
vertebrate mammal placental carnivore canine dog</thread>

<thread>entity object living_thing organism animal chordate
vertebrate mammal placental carnivore feline cat</thread>
```

Their lowest common meeting place is "carnivore." Next we count how much remains on the tails of both of the threads to get our total distance, which in this case is four—two on each thread. This distance metric is known as the distance from the "fork point" and is a convenient and accurate way to compare two words which

may not be at all alike. The closer they are in type, the lower down in the threads the fork will be.

In addition to comparing two words, it is necessary to be able to make one word more like another. Again, the threads aid this endeavor. One word is made more like another by traversing its thread up to the fork point and then down the other thread. As an example, if we wanted to make "cat" more like "dog", it would pass through "feline," then "carnivore," then "canine," then "dog."

4.2 Implementation

The project is implemented using over thirty Java classes, compiled in Java 5.0. A large part of the code base is devoted to the Swing user interface, which is extensive and much of which was written by Amanda Smith and Amanda Seybold. The rest of the code manages the SOMs and runs the conversation algorithms.

4.2.1 User Interface

It is both a strength and a weakness of Self-Organizing Maps that the only way to really understand the changes in them without running complicated and potentially misleading clustering algorithms, is to look at them with a human eye. This is simple if the nodes are colors, but quite complicated when a node can have any noun in the English language as its content, paired with a percentage value and a boolean for like or dislike. Thus, an extensive user interface was created in order to manipulate the model in a way meaningful to a human user.

The user interface displays "people" by showing their SOMs. There is a square on screen representing each node in the Map, and the square's color is dependent upon the trust value of the node and whether the content is liked or disliked. Blue indicates like and red dislike, and the saturation of the color reflects the trust value. Thus squares with very low trust values appear gray, indicating that the are close to the border between like and dislike. Mousing over the square pops up a tooltip that displays the content and the trust value, so that it requires very little effort to view

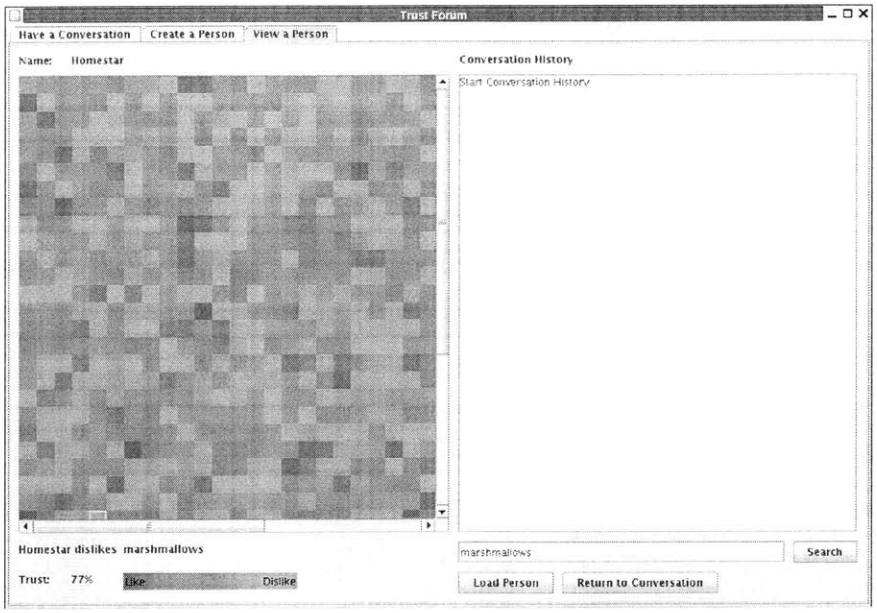


Figure 4-1: View and search interface

the content of any cell. Additionally a map can be searched for a specific topic.

New people can be created via the interface, by entering the three relevant pieces of information—topic, "like" or "dislike," and trust/belief percentage—for every cell. People can also be randomly generated using this interface, or some blend of the two methods.

People can be saved and loaded at any point in their "lives." Saving saves not only their SELF map, but their maps of all other people, and everything that has been said to them, in the order it was said and by whom. Loading restores all that information. In addition, the save format is human readable, printing out every node in a consistent order, so it doubles as a way to quickly convert current state to a form that can be read later. Later you see examples of SOMs presented in this format.

The cornerstone of the model—the ability for two "people" to talk to each other—is also interface-supported. The main screen allows two people to be loaded, and for each displays both the SELF map, and the subjective map of the other person. There are boxes to enter the content of the sentences spoken by each; after each sentence the program displays the primary node that changed (at each iteration between six and nine nodes change, but there is always one that is primary). A conversation can

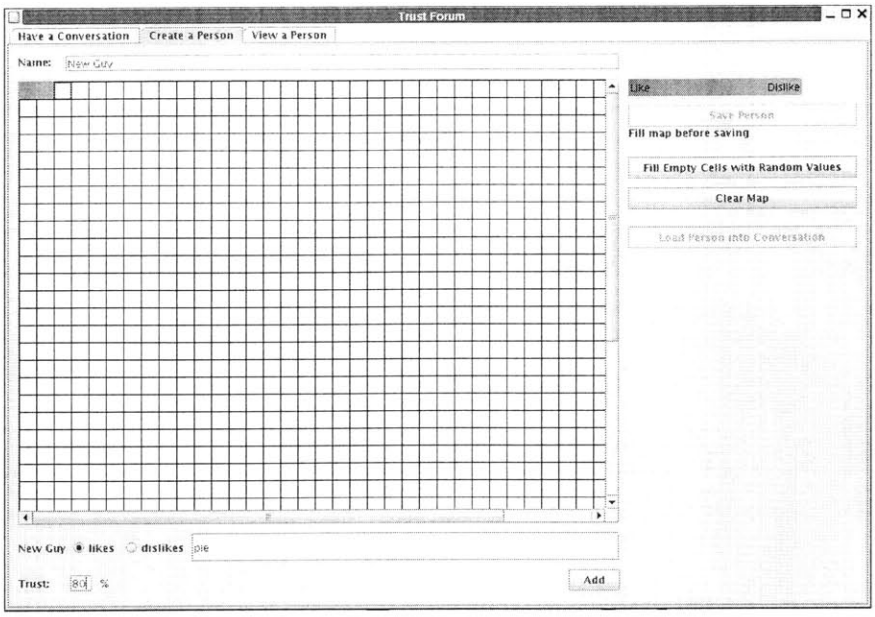


Figure 4-2: Person creation interface

also be saved in a simple text format and then loaded into the interface, allowing conversations to be generated via a script.

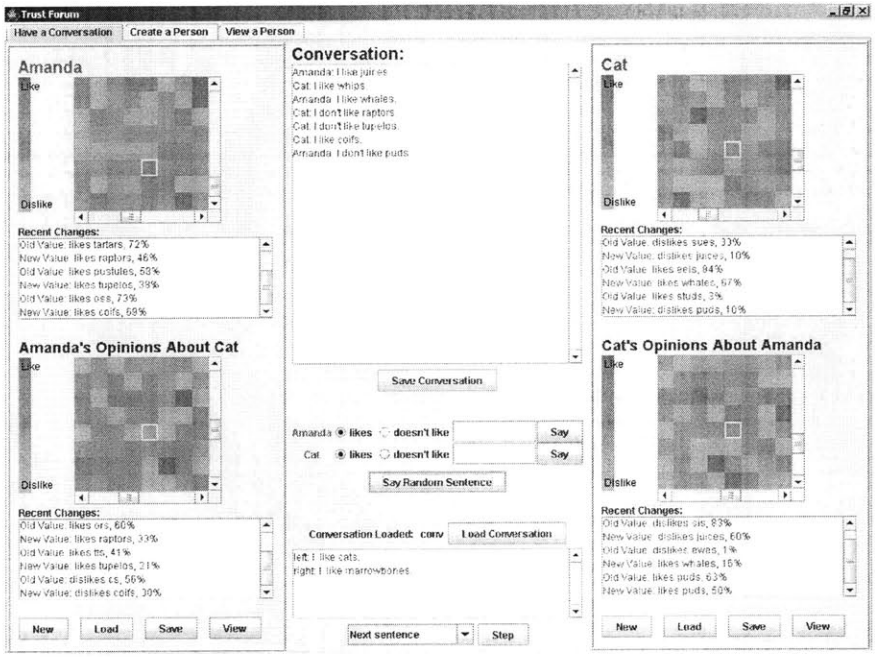


Figure 4-3: Conversation interface—the main screen for user interaction with the model. Two people are loaded, one on each side, and the spoken conversation appears in the window between them.

Chapter 5

Results

Running tests on large maps sizes with hundreds of disparate topics is easy to do, but hard to interpret. For that reason I will present the results of simple tests whose behavior is easier to see. That these results can be generalized to larger cases should be clear; we will consider a microcosm, but there is nothing different about the behavior because of it.

In one test there are two people, one of whom is Talker, whose likes and dislikes don't really matter, since his main purpose in life is to talk to our other person, Antigone. Antigone loves animals and hates cars. Her initial opinions of Talker are randomized, but are the same for every test. She looks like this:

```
Fri Apr 14 18:56:32 EDT 2006Antigone
0.0 0.0 0.9 dogs like
0.0 1.0 0.91 dogs like
0.0 2.0 0.9 dogs like
0.0 3.0 0.85 cats like
0.0 4.0 0.8 horses like
1.0 0.0 0.85 cats like
1.0 1.0 0.85 cats like
1.0 2.0 0.8 horses like
1.0 3.0 0.8 horses like
1.0 4.0 0.75 mice like
2.0 0.0 0.75 mice like
2.0 1.0 0.75 mice like
2.0 2.0 0.9 cars dislike
2.0 3.0 0.87 cars dislike
2.0 4.0 0.89 motors dislike
3.0 0.0 0.9 cars dislike
3.0 1.0 0.8 cars dislike
3.0 2.0 0.85 bikes dislike
3.0 3.0 0.92 bikes dislike
3.0 4.0 0.78 motors dislike
4.0 0.0 0.76 tires dislike
4.0 1.0 0.76 tires dislike
4.0 2.0 0.8 cars dislike
4.0 3.0 0.75 brakes dislike
4.0 4.0 0.8 brakes dislike
```

Figure 5-1: A list of each node in Antigone's SELF map listed. X and Y coordinates of each node are the first two numbers listed on each line; the third number is the belief level. This map was hand-coded to specifically talk only about animals and cars.

```
Fri Apr 14 18:57:17 EDT 2006Talker
0.0 0.0 0.8449560662913401 bodys dislike
0.0 1.0 0.12706835171195363 sauropods dislike
0.0 2.0 0.8686178225987744 copulations dislike
0.0 3.0 0.5343144217743439 clasts like
0.0 4.0 0.2369061102365756 tiers dislike
1.0 0.0 0.19590095582600875 tideways like
1.0 1.0 0.13142258099269466 geographers dislike
1.0 2.0 0.1493676194334339 shavians dislike
1.0 3.0 0.03532015726420412 maters like
1.0 4.0 0.4960353104872133 alternators like
2.0 0.0 0.8517391179068259 mimics like
2.0 1.0 0.15113948343466677 cricketers like
2.0 2.0 0.14315885670397677 planets like
2.0 3.0 0.06236750465080665 muslins dislike
2.0 4.0 0.6644249498010764 foundlings like
3.0 0.0 0.3571572137965756 effluents like
3.0 1.0 0.913969437546307 bearcats dislike
3.0 2.0 0.008997761190323916 teleprinters like
3.0 3.0 0.3857501120352197 marabous dislike
3.0 4.0 0.6491054628628916 elands like
4.0 0.0 0.5686919850642541 opossums dislike
4.0 1.0 0.09223112481051499 slops dislike
4.0 2.0 0.5109224375700278 beets like
4.0 3.0 0.9939876087360072 adoptees like
4.0 4.0 0.9428489881243048 trichloromethanes like
```

Figure 5-2: Antigone's map of Talker, as randomly generated initially.

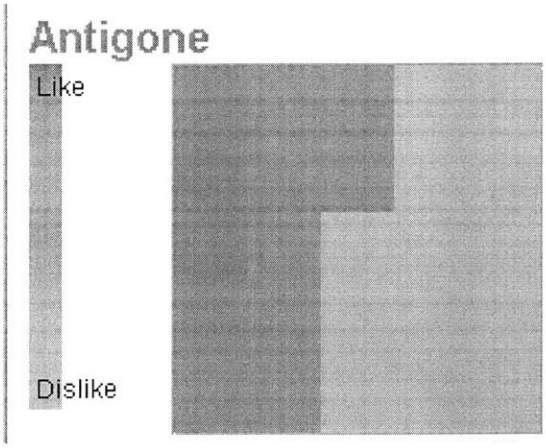


Figure 5-3: Graphical representation of Antigone's SELF map. Darker colors indicate like, in this case of animals. Lighter colors indicate dislike, in this case of cars.

The first number in a row corresponds to an x-coordinate and the second number to a y-coordinate. The third number is a "trust" value, the word following it is the topic in that map cell, and the last word is either "like" or "dislike" and is the opinion on that topic. The first set of rows is Antigone's SELF map, which I created by hand to have only two categories of objects in it. The second is her map of Talker, as randomly initialized. The random initialization takes words from WordNet, some of which can be obscure.

Antigone's graphical form appears above. Recall that darker sections indicate stronger like and lighter sections indicate stronger dislike. Thus the left side of the map is the portion that represents liking animals, and the right is disliking cars.

There are three different hand-coded mini-conversations I used for testing: Animal, Car, and Mix. In Animal, Talker says that he likes various animals—cats, dogs, and horses—in ten statements. In Car, Talker says that he likes various vehicle-related things—cars, brakes, and motors—in ten statements. Mix alternates statements about liking animals and liking cars for ten statements.

5.1 Untrustworthy Source Agrees With World View

What happens when Talker starts talking about how much he likes cats, given that the closest match to "cats" that Antigone knows about him is that he dislikes them, and she believes that very strongly? So when he begins to talk about liking animals, there is some contradiction: he is appealing to Antigone's world view, but at the same time being an untrustworthy person while doing so.

In the following test, Talker continuously says "I like cats." Antigone's level of belief that she likes cats varies as follows:

```
.85
.78
.78
.80
.82
.84
.86
.88
```

Her trust of Talker varies as follows. The word in quotations on each line indicates the best match word in her map of Talker at the time; it begins at "bearcats" and travels along the thread with each training iteration to arrive at "cat":

```
"bearcats" dislike .91
"civet" dislike .18
"viverrine" like .25
"carnivore" like .51
"cat" like .66
.76
.82
.87
```

So we see that Antigone's belief dips a little at first; someone untrustworthy saying that they like cats makes her like them less. Then as she comes to trust Talker more because he continues to be consistent and insistent on this point, her belief increases again, eventually passing its initial point.

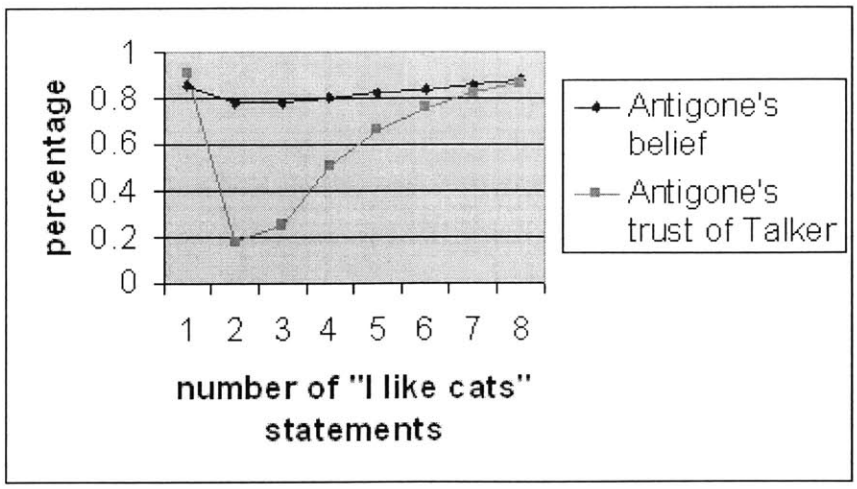


Figure 5-4: Graph of map behavior when Talker agrees with Antigone's current belief of animals, given that Talker starts as an untrusted source.

5.2 Priming the Listener

In this test Talker uses the Animal set to try to get Antigone to trust him by talking about things that she believes. Talker tries increase his ability to change Antigone's mind by "priming" her by saying things she agrees with. In this experiment Talker speaks the Animal set varying numbers of times before saying that he likes cars, and then we can see what the end result of Antigone's opinion of cars is.

If Talker begins talking about cars immediately, the closest match in Antigone's map of Talker is "likes teleprinters" at 1%. So Antigone has virtually no trust in Talker's opinion of them. If Talker initiates the conversation by saying "I like cars," then Antigone's opinion of cars will go from "dislike 87%" to "dislike 78%."

If Talker initiates conversation with the Animal set and then says "I like cars," her opinion goes from 87% to 69%. If he begins with two Animal sets, the change is from 80% to 63%. The starting percentage is lower because the changes caused by the Animal set have spilled over to cars. Beginning with three Animal sets has the same result. So clearly there is some priming effect, but there is a limit to the extent of it.

The limitations to the priming effect are largely due to the neighborhood of map

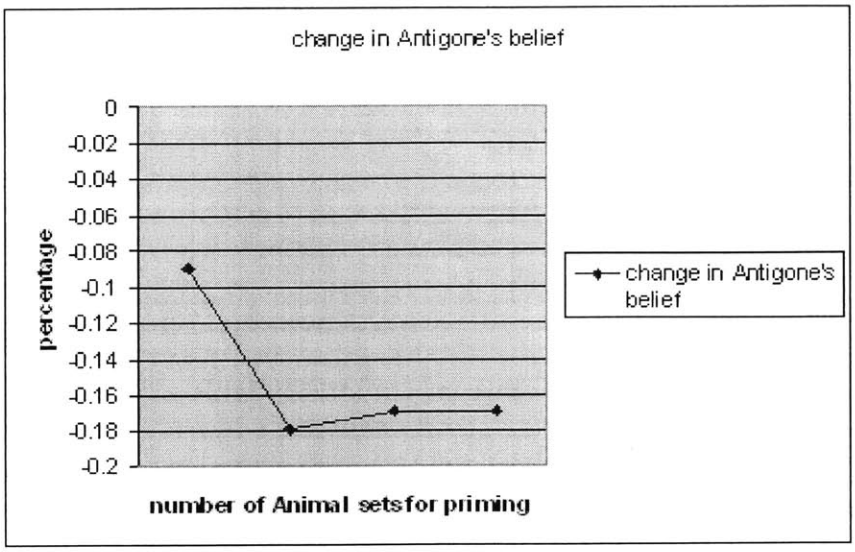


Figure 5-5: Graph of the change in Antigone's opinion of cars when Talker primes with either 0, 1, 2, or 3 Animal sets, and then says that he likes cars.

cells that are changed in each sentence. Each time Talker speaks a square region of nine cells is modified. That square of cells is then more likely to be a match to the next sentence, which keeps the changes confined within a small region, not spreading across the entire map.

5.3 Offending the Listener

In principle, the priming effect should have a converse, where deliberately disagreeing with a person causes them to believe everything less. In this test Talker will talk about his liking of cars before then saying that he likes cats (recalling that Antigone hates cars and likes cats). The Car set contains ten statements about liking cars.

As described before, if Talker begins the conversation with "I like cats," Antigone's opinion of cats goes from "like 85%" to "like 78%." If he talks for one or two Car sets and then says "I like cats," Antigone's opinion of cats stays constant at 85%. If he does three Car sets, and then says it, her opinion actually goes from "like 85%" to "like 86%."

This seems contradictory, and the reason for it is that talking about cars causes

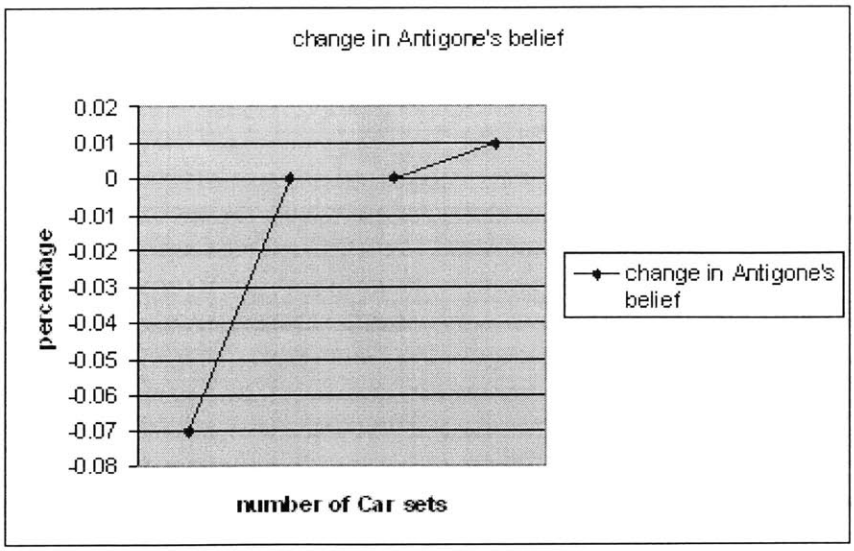


Figure 5-6: Graph of the change in Antigone's opinion of cats when Talker offends with either 0, 1, 2, or 3 Car sets, and then says that he likes cats.

some of the cells in Antigone's map of Talker to generalize. At the same time, it's internally consistent, so after ten statements Antigone is beginning to be convinced about cars, and gives more credence to Talker's opinions.

5.4 Self-Contradiction

If a person continually says contradictory things, this should reduce their trustworthiness. In this test Talker alternates saying that he likes and dislikes cats, and we take the average trust value in Antigone's map of Talker at the end.

After one set of ten statements of alternating like and dislike, the average trust in Antigone's map of Talker drops from 43.5% to 34.2%. After that the average fluctuates around 34% (the nodes actually changing vary between 15% and 20%). This is because Antigone's own opinion of cats causes her to give more weight to the statements of like than the ones of dislike, so the two don't cancel each other out. It still follows the general pattern that we don't trust people who are inconsistent.

Chapter 6

Discussion

The first thing to note is that there are very few parameters that need to be set. Even the equations themselves are robust against excessive modifications. Part of the goal of this thesis was to see what was possible with as few parameters as possible; hence, some of the following discussion will include problems that could be fixed by adding additional parameters.

6.1 Characteristics of Present Model

- Trust can be built up by catering to another person's world view. This is equivalent to telling someone what they want to hear, but because they have no prior knowledge of you, they believe it and think better of you for it.
- An untrustworthy source agreeing with you makes you believe something less. This may seem a little counterintuitive, but consider the 2004 presidential elections, when the rumor that "Saddam Hussein supports Kerry" was circulating. One tries to not be aligned with the untrustworthy source, and as a result questions existing beliefs
- When a person continually contradicts himself the result is a lack of trust. People with no internal consistency can't be trusted. The exact level to which they can't be depends upon the constants in the algorithm's equations.
- Order matters when it comes to conversations: saying something and then saying the opposite doesn't obliterate the first statement. Essentially, this means

that you can't erase the past, or take back something once said.

- As long as the conversation topics are somewhat broad-ranging, the "topic" fields of the SOMs become more general. One of the effects of this is that they are more likely to match new topics. This means that the longer people talk, not only do they begin to trust each other on the specific topics they are discussing, but that trust spreads further. When introducing a new topic it is more likely to be trusted because of this generalization.

- If you tell someone something enough, they begin to believe it. Each statement of like or dislike introduces more like and dislike into the universe, so eventually saying the same thing enough times will bring the other person in line with what you are saying. I think this is the theory behind brainwashing, but it's not clear how applicable it is to ordinary life.

- New topics obliterate the old. The map is a fixed size, and does not grow under this model, so there is a finite number of specific topics that can exist in it. So for example, if Antigone and Creon spend some time talking about tigers, and then talk about fish for awhile, the specific knowledge about tigers may get lost. This might be a reasonable memory model—that new information blots out the old—but there's no reason to choose any particular size of map. This essentially means that there is an additional and unintended parameter in the model: map size.

- Practically speaking, belief and trust values can never actually get to 100%. Because the equations involve a lot of averaging, it would take rounding error to ever believe something or trust someone completely. It could be argued that people never completely believe anything, but at the same time, I have a lot of confidence in the fact that I can not fly.

- The area of effect of one single topic of conversation or one single sentence is limited by the sphere of influence. Any individual sentence changes only nine map cells, therefore the largest effect of any one statement is to alter nine topics.

In principle, repeating the topic could cause the effect to alter another nine, for a total of eighteen changes, but the most likely event is that if the topic is repeated, the best match will be within the range of already altered cells. Thus the likely area of effect is only nine to fourteen cells. As the topic continues with minor variations more cells may get pulled in, but because of the likelihood that a match comes from the altered region, it is likely that only about sixteen cells will be altered from a single topic of conversation, no matter how many times it is repeated. This limits the effect on overall trust that one topic of conversation can have.

6.2 Speculation on Next Steps

- The equations are symmetric: trust is gained and lost at the same rate. There's no bias towards trusting people or distrusting people, which seems unlikely to correspond to reality. This could potentially be fixed by means of a "gullibility" parameter, but could be troublesome if psychologically speaking the rate of gaining trust has a different curve shape than the rate of losing it.

- If one wanted to represent the world—things that actually, physically happen—one could make a new person called World, and make all people trust World 100%. It would be necessary to prevent anyone from changing their opinion of World. It requires only a very small modification to add the ability for concrete actions to happen. Furthermore, insanity could be modeled by allowing someone's trust in World to change. Once World started contradicting what they believed, their trust in it would diminish and they'd stop believing the things they "see."

Chapter 7

Contributions

- Identified a tractable problem within the vast realm of providing context to memory.
- Designed a model of how humans learn to trust each other that is simple, with very few parameters. The model is extensible and customizable; the "characteristics" of a person are specified by a map and some constants in two equations. People can discuss their likes and dislikes across a wide range of topics; any topic with an annotated thread in WordNet will work.
- Implemented the model in Java, complete with graphical user interface.
- Tested the behavior of the model on several different types of conversations: an untrustworthy person agreeing with one's world view, a person continuously changing his mind, a person contradicting one's world view, and a person attempting to build trust by agreeing with one's world view.
- Showed that the tests reveal model behavior similar to what one might expect from a human. These simulated people can be swayed by propaganda, irritated by inconsistency, and brainwashed.

Bibliography

[1] Gary C. Borchardt. Causal reconstruction. AI Memo 1403, MIT Artificial Laboratory, February 1993.

[2] Ray Jackendoff. *Semantics and Cognition,* volume 8 of *Current Studies in Linguistics Series.* MIT Press, Cambridge, Massachusetts, 1983.

[3] Teuvo Kohonen. *Self-Organizing Maps.* Number 30 in Springer Series in Infomation Science. Springer, third edition, 2001.

[4] Stephen D. Larson. Intrinsic representation: Bootstrapping symbols from experience. Master's thesis, Massachusetts Institute of Technology, 2003.

[5] Marvin Minksy. K-lines: A Theory of Memory. AI Memo 516, MIT Artificial Intelligence Laboratory, June 1979.

[6] Marvin Minsky. *The Society of Mind.* Simon & Schuster, Inc, 1985.

[7] Seth Tardiff. Self-Organizing Event Maps. Master's thesis, Massachusetts Institute of Technology, 2004.

[8] Lucia M. Vaina and Richard D. Greenblatt. The use of thread memory in amnesic aphasia and concept learning. AI Working Paper 195, MIT Artificial Intelligence Laboratory, 1979.